Stochastic modelling and evaluation using GreatSPN

Elvio G. Amparore
Dipartimento di Informatica
Università degli Studi di Torino
Corso Svizzera 185, 10149, Torino, Italy
amparore@di.unito.it

ABSTRACT

GreatSPN is a tool that supports model-based (stochastic) analysis of Discrete Event Dynamic Systems (DEDS) modeled as Generalized Stochastic Petri Nets or one of its extensions like Stochastic Well-formed Nets, Deterministic and Stochastic Petri Nets among the other. Performance evaluation of the timed and stochastic properties of the modeled systems was the initial reason for the tool development, and it is today a large and flexible framework that incorporates several analysis techniques, performance index types, variegated transition timing specifications, etc. In this paper we report the current status of the GreatSPN framework, with a focus on the modularity, the types of stochastic analysis, the specification and evaluation functionalities, and its role for the performance evaluation.

Keywords

GreatSPN, performance evaluation, stochastic analysis, Petri nets.

1. INTRODUCTION

Modeling and evaluating the performances of a system requires a user to be capable of expressing the salient features of the system into an abstract form, to be capable of formulating analytical methods to compute relevant indices and infer useful properties and information. Therefore, a central part of the performance evaluation of system is the availability of abstract formalisms for modelling systems, and possibly the availability of software tools to automate the computation of analytical and numerical properties.

GreatSPN is a common framework of independent tools, developed starting from 1986, which is built around the Petri net model formalism. The framework incorporates more than 60 tools, with many integrated into a single common *Graphical User Interface* (GUI), and it is designed around the concept of a modelling workflow. Petri nets are used as the central model formalism due to their simplicity, and their intuitive way to model asynchronous and concurrent systems. Several extensions have been formulated to express timed and stochastic event definitions, most notably the *Generalized Stochastic Petri Net* (GSPN), which is an excellent formalism for performance evaluation of large systems due to its automatic conversion into the underlying *Continuous-Time Markov Chain* (CTMC). Two important extensions to the GSPN are included: *Stochastic Well-formed Nets* (SWN), which increase the power of the modelling formalism, and *Deterministic and Stochastic Petri Nets* (DSPN), which allow rich characterization of the firing time distributions.

The GUI plays an important role, since the most important tools are accessible in a user-friendly way, while providing a simple environment to perform all the basic operations (modeling, verification, analysis, simulation, etc).

In this paper a description of the current status of GreatSPN is provided, starting from an overview of the main tools, the kinds of stochastic analysis available, who is the target user base for this framework. A brief example of a modeled system is also provided, together with reference to recent case studies performed with GreatSPN. GreatSPN is open source, and links on how to obtain the tool are included at the end.

1.1 Comparison with other Petri net tools

Several other tools are built around similar concepts of GreatSPN, but only few provide a comparable platform in terms of tools, model formalisms, numerical methods, integration and user-friendliness. A surely incomplete list follows.

TimeNet. Derived from DSPNExpress (originally inspired by GreatSPN), TimeNET [19] is focused on steady state and transient numerical solutions of DSPN systems, and simulation. For the latter, TimeNET provides very advanced techniques as well as a flexible way to express general firing time distributions.

APNNtoolbox. Based on a different flavour of stochastic colored Petri nets known as *Abstract Petri Net Notation* (APNN), the tool [9] supports the generation of large reachability graph using Kronecker representation for the matrix encoding, and their efficient solution in steady state.

CPN-Tools. It is a tool [12] to edit, simulate and analyse Petri nets designed in a highly flexible colored extension, the *Colored Petri Net* (CPN). However, the tool does not use stochastic timings for the transition firings, limiting the use cases that can be modeled and analyzed.

Möbius. The tool Möbius [11] is a dependability and performance modelling environment designed around a different extension of the Petri net formalism, the *Stochastic Activity Network* (SAN). Möbius is probably the most mature tool for Petri nets, and supports a wide range of numerical and simulation tools.

Oris. The tool Oris [16] is designed around timed and stochas-

TOSME2021 – Tools for stochastic modelling and evaluation

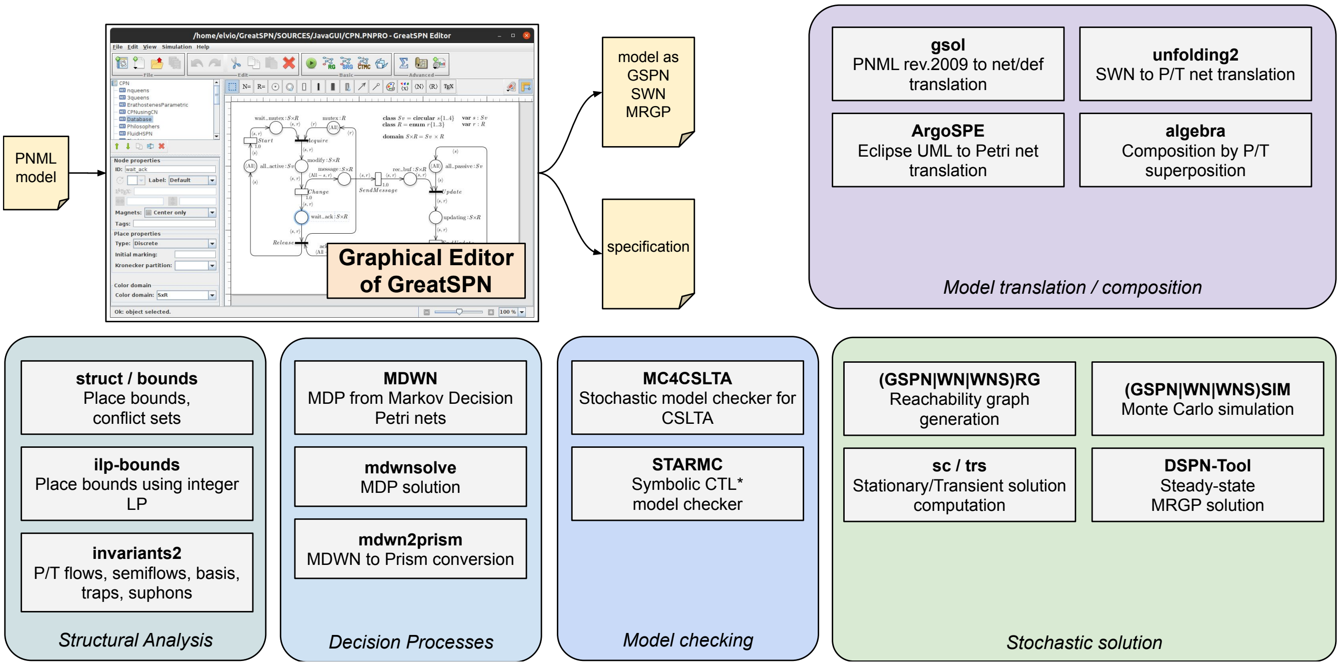


Figure 1: Overview of the GreatSPN framework.

tic Petri nets, with a focus on the analysis of non-Markovian system (more general than DSPN), for which it is the most advanced tool available today.

PIPE. The *Platform Independent Petri Net Editor* tool, or PIPE [8], is a Java-based application with many features for modeling and analyzing Petri nets. PIPE offers a nice interactive editor, animated token game, structural analysis, state space construction, stochastic analysis and simulation. It has several advantages, including being very compact and portable across multiple platforms.

JMT. The *Java Modeling Tools* [7] is a framework for performance evaluation for queueing Petri net models. It provides simulation as well as analytical solution tools, including an implementation of the Mean Value Analysis (MVA) algorithm (which is missing in the GreatSPN framework), asymptotic bound analysis, and workload analysis.

QPME The *Queueing Petri net Modeling Environment* [14] is an Eclipse plugin for modeling and analyzing hyerarchical colored queueing Petri net models. It features a graphical editor and an advanced simulation engine.

2. PERFORMANCE EVALUATION USING GREATSPN

The GreatSPN framework is a collection of tools that can be roughly organized by their functions, into these tool categories:

1. **GUI** The unified graphical interface [1];
2. **Translation** and composition tools to deal with multiple formats and model unfolding;
3. **Structural** analysis of model properties;
4. **Markov Decision** processes support (MDP);
5. **Model Checking** of temporal logics;
6. **Stochastic solution** using the Reachability Graph (RG), simulation, or differential equations.

Figure 1 depicts the main tools of the framework, organized by their logic functions. In this short tool paper we will focus on the stochastic solution module, which comprises a collection of tools to deal with numerical and simulation-based analysis.

2.1 Input formats

GreatSPN has been recently modernized to support the PNML [13] standard format (both P/T and Symmetric Nets). Unfortunately, PNML does not currently have a proper way to encode stochastic information on the net elements, hence only the net structure can be imported/exported. For the remaining purposes, the main input format remains the non-standard but broadly supported net/def format. Other exporters (APNN, GrML, NetLogo) are also available.

2.2 Type of stochastic analysis

The core concept of the framework is designed around an integrated workflow for modeling and verification. Figure 2 illustrates the core steps of the workflow. The framework user first draws the model using the Petri net formalism. Models can be modular, i.e. they are made of separate independent nets that are combined together using net composition algebraic operators. The workflow follows these steps:

1. Draw the Petri net model, usually with one of the supported extensions (GSPN, SWN, DSPN).
2. Verify the model structure by means of its structural properties, which include: P/T-invariants, P/T-flows,

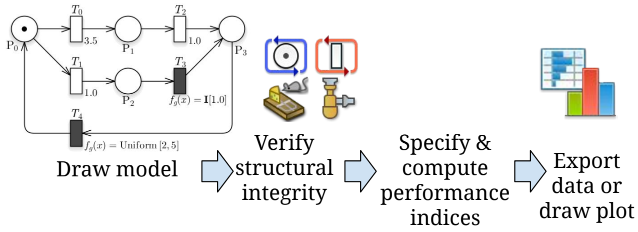


Figure 2: Workflow of the GreatSPN framework.

siphons, traps, interactive token game simulation, deadlock analysis, CTL(continuous time logic)/LTL(linear time logic)/CTL* properties, etc.

3. Specify a tool and a set of target measures to compute with, and run the tool interactively from the GUI, or manually from the command line.

4. Export the resulting plot or data in a common format.

All tools exist as command-line binaries, which allows to automatize batch and complex job runs.

2.3 Relevant techniques

2.3.1 steady state and transient numerical solutions

The basic stochastic solution employs the construction of the *Reachability Graph* (RG) of the Petri net model. There are several flavours of RG that can be built:

1. **GSPNRG**: basic RG of GSPN models.

2. **WNRG**: RG of SWN models, i.e. supporting colored transition firings.

3. **WNSRG**: Symbolic RG of SWN models, exploiting model symmetries.

4. **DSPN-Tool**: RG of Markov Regenerative Processes (MRgP) of P/T models using general transition firings.

5. **STARMC**: RG encoded using Decision Diagrams [5]. Only limited stochastic solutions (steady state with Jacobi method) are possible, but very large state spaces (10^{100} and beyond) can be encoded.

Upon state space construction, a numerical solution can be performed, in steady state or in the transient state at a specific time t. DSPN-Tool and STARMC only support steady state solutions.

Performance indices are expressed using a domain-specific language that allows to formulate expressions using place marking probabilities, place distributions or transition throughputs. Global place marking distributions and transition throughputs can also be exported in CSV and Excel format, to be processed with external tools.

2.3.2 Stochastic model checking using CSL^{TA}

A more advanced dependability approach is the CSL^{TA} logic (Continuous Stochastic Logic with Timed Automata), in which the computed performance index is expressed using a Deterministic Timed Automata (DTA). The DTA can be designed graphically in the GreatSPN GUI, and then used to compute indices on models. The DTA can contain *clocks* and *clock resets*, which allow for complex timing specifications.

2.3.3 Solution of DSPN systems

The DSPN solver deals with an extended Petri net model formalism, which includes marking parameters, rate parameters, predefined transition policy (a transition may be operated by multiple parallel servers, or its delay may be defined by a marking-dependent function), general distribution firing times expressed as probability density functions, marking-dependent arc functions, and transition guards. This solver can either treat CTMC (in steady state and in transient) and MRgP (only in steady state) and employs advanced numerical solution methods (Krylov-subspace methods). The RG size that can be treated is in the order of the tens of millions of states. See [2] for a comparison with the other existing DSPN tools.

2.3.4 Simulation

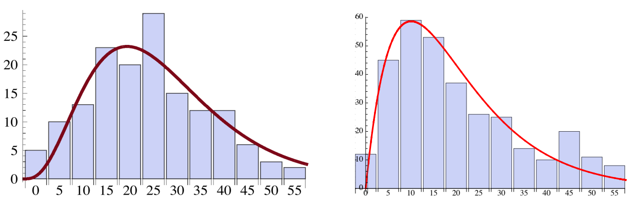
A Monte Carlo simulator is available in the framework, and can deal with solutions of models using batch simulation in transient or in steady state. The simulator supports SPN models with an arbitrary number of transitions with general firing time distributions (uniform, Erlang, etc.). A comparison involving the use of the GreatSPN simulator can be found in [18], where the high accuracy and the performance of the algorithms employed by the simulator is shown. The tool is guided by the target performance indices to compute, and iterates over produced events until all indices are below the accuracy threshold.

3. TOOL TARGETS

While the GreatSPN framework has been used in multiple contexts (biological modeling [17], industrial case studies [6], networking [15], to name a few) in the course of its long history, the framework itself was designed as a tool for research. In the last years, a special effort was made to reorganize the tools and the interface to be easy to use for teaching purposes [4]. The reorganization is done by separating basic/intermediate/advanced features, in order to provide a more gradual learning curve to the students learning hot to use the framework.

3.1 Recent GreatSPN case study

We summarize briefly a recent case study [3] of an Italian multi-utility company that operates in a geographical area of about 2200 km^2, with 531k inhabitants. The company manages water and gas distribution networks. To comply with national regulations, the company is required to intervene in case of a suspected gas leakage with a technician on-site in less than an hour, for at least 95% of the client calls.



(a) Travelling time distribution. (b) Securing time distribution

Figure 3: Fitted data distributions.

Figure 3 shows the available data for the travelling times and securing times of the operators, extracted from a company log and used to fit the model distributions. The remaining model parameters were derived from interviews and discussions with the domain experts.

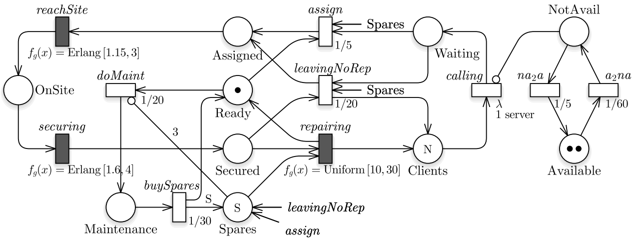


Figure 4: Parametric model of the multi-utility company.

Figure 4 shows the DSPN model used to study the parameter optimization problem. The results showed that with the expected inter-arrival time of the client's requests of about 30 minutes, four technicians are needed to reach the regulatory targets. Full details can be found in the original paper.

3.2 Recent GreatSPN features

While GreatSPN is an old framework, several new additions and improvements were added in the last few years. Recent features include, among others:

- The model checking STARMC that is based on multi-valued decision diagrams [5] for state space encoding.
- The new Java-based GUI, that replaced the previous, outdated one and which integrates a modern design workflow [1].
- Support for the standard PNML format, both in input and in output. The original custom model format (known as *net/def*) is still used internally between the various tools.
- GreatMod: platform for System Biology models [10].
- The DSPN solver [2], presented in the short case study in Section 3.1.

3.3 Tool availability and distribution license

GreatSPN is open source (GPLv2) and it is available on all major platforms (Linux, Windows, MacOS). Sources can be downloaded from the project's GitHub repository[1], along with the instructions to build the framework. Currently, building GreatSPN requires to build its two main dependencies first, Spot[2] and Meddly[3]. To ease the installation, a VirtualBox/VMware image is also available[4], or in the form of a *Dockerfile* to be used in an isolated container with Docker. A special version of the framework tailored for modeling and studying biological systems and integrated with the R language is also available [10].

[1] `https://github.com/greatspn/SOURCES`
[2] `https://spot.lrde.epita.fr/`
[3] `https://github.com/asminer/meddly`
[4] `http://www.di.unito.it/~greatspn/VBox/`

4. CONCLUSIONS

GreatSPN is a feature-rich framework, with a modern integrated graphical user interface, that allows to model Petri net models with a rich extended language, study their structural properties, define and compute performance indices. The framework consists of more than 60 tools that have been thoroughly used and tested. It works on all major modern platforms, and its source code is open source. The recent interface redesign has been devoted to simplify the interface for students, newcomers and practitioners, while still retaining a flexible and scriptable modular structure for research and development purposes. It is still actively developed, with new advanced functionalities still being added to the core modules.

5. REFERENCES

[1] E. G. Amparore. A new GreatSPN GUI for GSPN editing and CSL TA model checking. In *International Conference on Quantitative Evaluation of Systems*, pages 170–173. Springer, 2014.

[2] E. G. Amparore and S. Donatelli. DSPN-Tool: a new DSPN and GSPN solver for GreatSPN. In *2010 Seventh International Conference on the Quantitative Evaluation of Systems*, pages 79–80. IEEE, 2010.

[3] E. G. Amparore and S. Donatelli. alphafactory: a tool for generating the alpha factors of general distributions. In *International Conference on Quantitative Evaluation of Systems*, pages 36–51. Springer, Cham, 2017.

[4] E. G. Amparore and S. Donatelli. GreatTeach: a tool for teaching (stochastic) Petri nets. In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 416–425. Springer, Cham, 2018.

[5] J. Babar, M. Beccuti, S. Donatelli, and A. Miner. GreatSPN enhanced with decision diagram data structures. In *International Conference on Applications and Theory of Petri Nets*, pages 308–317. Springer, 2010.

[6] S. Bernardi, S. Donatelli, and A. Horvath. Compositionality in the GreatSPN tool and its application to the modelling of industrial applications. In *University of Aarhus (Denmark*. Citeseer, 2000.

[7] M. Bertoli, G. Casale, and G. Serazzi. JMT: performance engineering tools for system modeling. *SIGMETRICS Perform. Eval. Rev.*, 36(4):10–15, 2009.

[8] P. Bonet, M. Catalina, and R. Puigjaner. A petri net tool for performance modelling. *PIPE (CLEI)*, 2(5), 2007.

[9] P. Buchholz and P. Kemper. A toolbox for the analysis of discrete event dynamic systems. In *International Conference on Computer Aided Verification*, pages 483–486. Springer, 1999.

[10] P. Castagno, S. Pernice, G. Ghetti, M. Povero, L. Pradelli, D. Paolotti, G. Balbo, M. Sereno, and M. Beccuti. A computational framework for modeling and studying pertussis epidemiology and vaccination. *BMC bioinformatics*, 21(8):1–32, 2020.

[11] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. Webster. The mobius modeling tool. In *Proceedings*

9th International Workshop on Petri Nets and Performance Models, pages 241–250. IEEE, 2001.

[12] K. Jensen and L. M. Kristensen. *Coloured Petri nets: modelling and validation of concurrent systems.* Springer Science & Business Media, 2009.

[13] E. Kindler. The Petri Net Markup Language and ISO/IEC 15909-2: Concepts, status, and future directions. *Entwurf komplexer Automatisierungssysteme*, 9:35–55, 2006.

[14] S. Kounev and C. Dutz. QPME: a performance modeling tool based on queueing Petri nets. *SIGMETRICS Perform. Evaluation Rev.*, 36(4):46–51, 2009.

[15] R. Lai. Performance results for the CSMA/CD protocol using GreatSPN. *Journal of Systems and Software*, 37(1):75–90, 1997.

[16] M. Paolieri, M. Biagi, L. Carnevali, and E. Vicario. The ORIS tool: quantitative evaluation of non-markovian systems. *IEEE Transactions on Software Engineering*, 2019.

[17] S. Pernice, M. Beccuti, G. Romano, M. Pennisi, A. Maglione, S. Cutrupi, F. Pappalardo, L. Capra, G. Franceschinis, M. De Pierro, et al. Multiple Sclerosis disease: a computational approach for investigating its drug interactions. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 299–308. Springer, 2019.

[18] R. J. Rodríguez, S. Bernardi, and A. Zimmermann. An evaluation framework for comparative analysis of generalized stochastic Petri net simulation techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(8):2834–2844, 2018.

[19] A. Zimmermann. Modelling and performance evaluation with timenet 4.4. In *International conference on quantitative evaluation of systems*, pages 300–303. Springer, 2017.